IRIS: VISUAL-SEMANTIC BINDING FOR DIFFUSION IMAGE WATERMARKING

Anonymous authors
Paper under double-blind review

ABSTRACT

In-generation watermarking provides a promising approach to tracing diffusion-generated images, but recent work shows such watermarks can be transplanted onto arbitrary images because they are agnostic to the visual content being watermarked. Prior content-aware methods bind the watermark only to prompt-level semantics, leaving it loosely coupled to the visual content it watermarks. We present IRIS, a training-free in-generation watermarking framework that binds each watermark to the visual semantics of the generated image. This introduces two challenges: (1) deriving the watermark requires the visual semantics of the generated image, yet embedding that watermark during generation may itself alter those semantics, creating a circular dependency, and (2) verification should be sensitive to semantic changes while resilient to common image distortions. To address the first challenge, IRIS derives the watermark from the visual semantics of a preview. It embeds the watermark as a small perturbation in the Fourier domain of the latent only during the final few sampling steps, thereby preserving those semantics. To address the second challenge, IRIS constructs distortion-robust semantic codes from augmented visual representations. At detection, IRIS derives the expected watermark from the query image and verifies it against the inverted latents. Extensive experiments show that IRIS reduces the mean black-box forgery attack success rate from over 96% for content-agnostic methods and 11–61% for prior content-aware methods to 2.1%, while keeping high detectability and generalizing to SD3.5 and FLUX.2.

1 INTRODUCTION

Latent diffusion models (LDMs) enable high-quality, large-scale image generation for general purposes, powering widely used models such as Stable Diffusion (Rombach et al., 2022) and DALL-E 3 (Betker et al., 2023). This capability raises growing concerns about the governance of AI, including its potential misuse for producing unsafe content and disinformation, and also the copyright of generated images. In response, watermarking has emerged as a promising technique for responsible AI by enabling source attribution of AI-generated content and has been adopted in commercially deployed generative AI systems (OpenAI, 2026; Anthropic, 2026) in response to the latest requirements for content provenance and regulatory compliance (European Union, 2024; C2PA, 2026). For diffusion-generated images, in-generation watermarking has become a prominent approach that embeds watermarks into latents during diffusion generation, such that generated images carry watermarks from the moment they are created, with the watermark being imperceptible and robust (Wen et al., 2023; Yang et al., 2024b; Gunn et al., 2025; Zhang et al., 2026; Yang et al., 2026).

However, recent studies indicate that existing in-generation watermarking methods are far from desirable. They leave traces in the latents, allowing attackers to distill a residual pattern by averaging watermarked images and inject it into arbitrary images, causing them to be detected as watermarked (Yang et al., 2024a), or transplant the watermark of a single queried watermarked image to an arbitrary image via a reprompting attack and an imprint attack (Müller et al., 2025). These attacks make the watermark falsely authenticate a forged image. Existing in-generation watermarking methods are vulnerable to these forgery attacks for a shared reason. Their watermarks are agnostic to the content of the images they aim to authenticate (Yang et al., 2024a). Therefore, binding the watermark to image semantics enables it to resist these forgery attacks.

Reviewers: please read the Reviewer Guidelines (iclr.cc/Conferences/2027/ReviewerGuidelines) and the AI Policy for Reviewers (iclr.cc/Conferences/2027/AIPolicyForReviewers). If you used AI to expand, edit, or polish your review, please provide the input text to the LLM. Better still, consider skipping the LLM and submitting your original text: we, and the authors, are much more interested in your unedited thoughts than in what an LLM has to say. AI-assisted or not, you are putting your name and reputation behind your review: LLM-generated falsehoods, hallucinations or misrepresentations are subject to disciplinary action, which may include desk-rejecting all papers you have authored.

Recent efforts (Arabi et al., 2025; Zhang et al., 2026) have attempted to bind the watermark to the semantics of the generated image, but they do so at the caption or prompt level. We argue this is not merely a coarser choice but a workaround for a circularity: these methods embed the watermark into the *initial* noise, yet the initial noise determines the generated image whose semantics the watermark is meant to be bound. The prompt is the only image-related quantity available before generation starts, so the binding is forced to stop there—and the final generated image carrying the watermark is visually distinct from the auxiliary image from which the watermark was derived. Images from the same prompt differ by 55° in CLIP space on average (Sec. 4.1), leaving room to exploit for watermark transfer.

Our goal is to break this circularity and bind the watermark to the *visual* semantics of the image it authenticates, without leaving room for exploiting watermark transplantation. Doing so raises two challenges: (i) *the watermark embedding should not alter the visual semantics of the generated image it is conditioned on*. To address this, IRIS derives the watermark from an unwatermarked preview, restarts at a late sampling step where the latent has largely converged, and embeds the watermark as a small mid-frequency Fourier perturbation such that the final image remains visually identical to the content from which the watermark is derived. (ii) *the watermark verification should be sensitive to visual-semantic changes yet resilient to common distortions*. To address this, IRIS maps a semantic code from visual representations in canonicalized and augmented views with a local candidate set and a ratio-based correlation metric for detection.

Our contributions lie in three aspects:

1. **A novel watermarking framework**. We develop IRIS, a training-free, in-generation watermark that preserves the visual semantics of the generated image while embedding a watermark derived from those semantics, achieving a tight binding that is sensitive to visual semantic changes in watermarked images during detection.
2. **Extensive Experiments**. We conduct empirical studies that motivate the key design choices of IRIS, followed by comprehensive evaluations against both content-agnostic and content-aware methods, showing that IRIS significantly reduces forgery success rates while preserving fidelity.
3. **Key Insights**. Detailed analysis and ablation studies provide insights into the sensitivity of visual-semantic binding, the effectiveness of detection statistics for forgery resistance, and the feasibility of deploying IRIS dual watermarks with initial noise-based methods.

2 Related Work

Latent Diffusion Models.

Diffusion models are trained to learn the reverse denoising process from noise (Ho et al., 2020). A latent diffusion model generates an image by progressively denoising Gaussian noise in the latent space and subsequently decoding the final latent into an image (Rombach et al., 2022). For in-generation watermarking, a key property of diffusion models is that, with a deterministic sampler, their generation trajectory can be deterministically inverted. DDIM is such a widely used deterministic sampler (Song et al., 2021). Specifically, given an image generated by an LDM with DDIM sampling, DDIM inversion can approximately reconstruct the intermediate latents at each denoising step, including the initial Gaussian noise. DDIM inversion was firstly adopted for in-generation watermarking by Wen et al. (2023) and followed by subsequent works (Ci et al., 2024; Huang et al., 2024; Li et al., 2025; Zhang et al., 2026). Another important component of LDMs that affects watermarking performance is the variational autoencoder (VAE). Since the watermark is embedded in the latent space rather than in pixel space, more precise transformations between the latent and image spaces (Hong et al., 2024) can improve watermark detectability, but may incur additional computational overhead (Gunn et al., 2025). This inversion-based paradigm can also be extended to newer flow matching models, which predict a velocity field and generate samples by solving an ODE (Lipman et al., 2023; Esser et al., 2024). Throughout this paper, we use *diffusion models* as an umbrella term for both diffusion-based and flow-matching-based generative models.

In-Generation Watermarking for Diffusion Images.

Depending on when a watermark is embedded, watermarking techniques for diffusion-generated images fall into three categories: post-hoc, decoder-based, and in-generation. Post-hoc methods

embed watermarks after image generation (Cox et al., 2007; Zhu et al., 2018). Decoder-based methods use a fine-tuned VAE decoder to embed watermarks (Fernandez et al., 2023). In contrast, in-generation watermarking embeds watermarks in the initial noise or latents of the denoising process, a prominent paradigm due to its strong robustness (Zhao et al., 2024) and great image quality. Tree-Ring (Wen et al., 2023) writes a pattern into a low-frequency annular region of the Fourier domain of the initial noise; RingID (Ci et al., 2024) extends it with multi-channel patterns and multi-bit capacity. Gaussian Shading (Yang et al., 2024b) maps a secret message to a Gaussian-preserving latent. ROBIN (Huang et al., 2024) locates the pattern to an intermediate latent with a hidden guiding prompt. GaussMarker (Li et al., 2025) jointly encodes an initial noise and embeds a Fourier-domain pattern into it. PRC-Watermark (Gunn et al., 2025) seeds the initial noise from a pseudorandom error-correcting code. T2SMark (Yang et al., 2026) uses Tail-Truncated Sampling to construct initial noises as watermarks. However, these methods embed watermarks agnostic to the content of the generated image they authenticate, leaving a one-to-many mapping between watermarks and images that attackers can exploit to forge the watermark (Müller et al., 2025).

Black-Box Forgery and Content-Aware Watermarking.

Many works exploit the vulnerability of black-box forgery of in-generation watermarking that is agnostic to the content of the generated images. Yang et al. (2024a) demonstrate how averaging a collection of content-agnostic watermarked images could reveal the underlying watermark pattern. Müller et al. (2025) imprints a targeted watermark into real images by manipulating the latent of an arbitrary image in an unrelated LDM to get closer to the latent of a watermarked image, and generates images with the target watermark by inverting a watermarked image and re-generating it with an arbitrary prompt. Jain et al. (2025) forge a watermark via a single watermarked image by introducing perturbations to the images. Recent work attempted to resist these forgeries through content-aware watermarking. SEAL (Arabi et al., 2025) embeds semantic information about the generated image into the initial noise, where the semantic information is derived from an auxiliary image's caption. SemBind (Zhang et al., 2026) uses a learned semantic masker yielding near-invariant codes for the same prompt and near-orthogonal codes across prompts to modulate a target latent before any standard latent-based watermark. However, these watermarks are not bound to the visual semantics of the images they authenticate, making their binding loose and leaving room for exploitation. In contrast, IRIS embeds a watermark that is more sensitive to visual-semantic changes by tightly binding the watermark to the generated image it authenticates.

3 Problem Statement

Notation. An LDM generates an image by sampling an initial Gaussian latent $z \sim \mathcal{N}(0, I)$ and iteratively denoising it, after which the final latent is decoded into an image x by a VAE decoder $\mathcal{D}$. We index the discrete steps of a sampling algorithm (e.g., DDIM (Song et al., 2021)) by $i = 0, \ldots, T-1$ in the order of generation, where $i = 0$ denotes the first sampling step and $i = T-1$ the last; these steps are distinct from, but correspond to the underlying (model-defined) diffusion timesteps. At detection, we run DDIM inversion from the encoded query image and index inversion steps by $t = 1, \ldots, D$, where $t = 1$ is the first step away from the image; $\hat{z}_t$ denotes the *inverted latent* at step t. We write Z_i and $\hat{Z}_t$ for the 2D Fourier coefficients of the watermarked latent channel of z_i and $\hat{z}_t$, respectively.

Problem setting. We consider an in-generation watermarking setting where the model owner aims to attribute generated images while preserving generation quality, resisting common image distortions, and resisting watermark forgery. This setting involves three sequential stages:

- *In Generation*: A model owner possesses a diffusion model and applies an algorithm with a secret key to generate an image x containing a watermark, which is then released to a user through an API. The watermark should be imperceptible to the user and preserve the generated image quality.
- *After Generation*: An image may undergo common image distortions during transmission, which may weaken its watermark. An adversary who can query watermarked images but has no access to the model or the secret key may attempt to falsely attribute a harmful image $\check{x}$ to the model owner by transferring the watermark of x to $\check{x}$.
- *In Detection*: Given a query image, the model owner tries to determine whether it contains a valid watermark attributing it to the model owner. The owner has access to the generative model and the secret watermark key, but does not require the text used for generation.

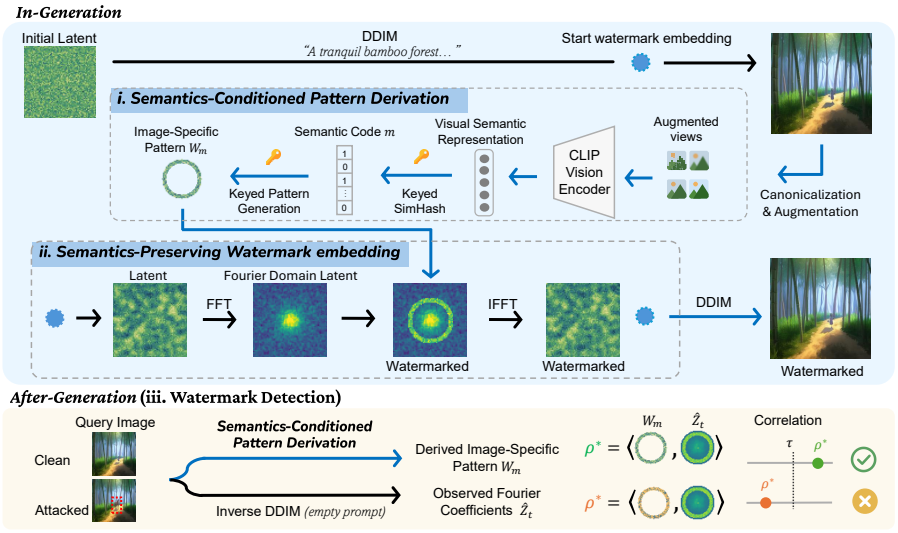

Figure 1: The IRIS framework.

4 METHOD

Overview of the IRIS framework. IRIS is a training-free, in-generation watermarking scheme whose verification is sensitive to visual-semantic changes by embedding an image-specific watermark derived from the visual semantics of the generated image and verifying the watermark through the visual semantics of the query image. As illustrated in Fig. 1, IRIS comprises three stages 1: (i) *Semantics-Conditioned Pattern Derivation*: derives an image-specific watermark from augmented visual representations via keyed SimHash and hashing; (ii) *Semantics-Preserving Watermark Embedding*: embeds the watermark without altering the visual semantics of the generated image while enabling detection; (iii) *Watermark Detection*: verifies an expected pattern derived from the query image against an observed signal in its inverted latent.

4.1 SEMANTICS-CONDITIONED PATTERN DERIVATION

To condition the watermark on the visual semantics of the generated image, IRIS first maps the representation of augmented clean previews to a resilient semantic code via SimHash (Charikar, 2002), a random-hyperplane-based locality-sensitive hashing (LSH) that maps high-dimensional representations to binary codes while preserving the similarity relationships of the original representations as much as possible. It then uses the semantic code and a secret key to generate a sequence of pseudorandom values. These numbers serve as the phases of the Fourier coefficients within an annular band, forming the watermark pattern.

Robust visual representation. In many real-world scenarios, watermarked images may undergo common distortions such as JPEG compression before verification. To minimize their impact on visual semantic code mapping, we generate K augmented views with random JPEG compression, additive noise, and blur, which then pass a defined canonicalization: (i) normalize the mean luminance, (ii) apply a heavy Gaussian blur, and (iii) apply heavy JPEG compression. The resulting representation $\bar{\mathbf{e}}$ is averaged over those of the K augmented and canonicalized views. We apply this preprocessing at *both embedding and detection* stages to absorb the impact of common image distortions for resilient verification.

Visual semantic codes. Given the visual representation $\bar{\mathbf{e}}$, we derive a bit string $\mathbf{m} \in \{0,1\}^\ell$ using a keyed *SimHash*:

$$m_j = \mathbb{I}\big(\mathbf{r}_j^\top (\bar{\mathbf{e}} - \boldsymbol{\mu}) \geq 0\big), \qquad j = 1, \ldots, \ell, \tag{1}$$

where $\boldsymbol{\mu}$ is the mean representation of a reference image set obtained using the same preprocessing and is kept fixed at both embedding and detection, and $\mathbf{r}_j$ is the j-th random hyperplane normal deterministically generated from the secret key. For two centred representations with angle α, each pair of corresponding bits differs with probability α/π, resulting in an expected Hamming distance

of $\ell\alpha/\pi$ bits between their codes. We empirically examine this relationship using CLIP representations (Radford et al., 2021) with $\ell = 24$. Images generated from *different prompts* have an average angular distance of $88.5°$, with expected and measured Hamming distances of 11.8 and 11.9 bits. For images from the *same prompt* with different seeds, these values decrease to $55.3°$, 7.4, and 7.6, respectively. Common image distortions cause much smaller changes: JPEG (Q=25) and Gaussian blur (σ=2) produce angular distances of only $7.9°$ and $4.7°$, corresponding to Hamming distances of 1.1 and 0.6, respectively.

Image-Specific Watermark. A watermark is a sequence of values embedded into a corresponding sequence of values in a carrier. In IRIS, we derive these values from the visual semantic code m using a keyed hash, resulting in an image-specific watermark. We embed the watermark in the Fourier domain of a latent channel. Fourier-domain watermarking has long been explored for digital images (Solachidis & Pitas, 2001) and has recently been adopted for in-generation watermarking of diffusion models (Wen et al., 2023). Following prior work, we place the watermark pattern in a ring-shaped region of the Fourier domain, which controls the frequency components affected by watermarking. We define a mask $M(\omega) = \mathbb{I}[\omega \in \mathcal{R}]$ for this region, where $\mathcal{R} = \omega : r_{\text{lo}} < |\omega| \leq r_{\text{hi}}$, ω denotes a Fourier-domain location, and r_{lo} and r_{hi} are the inner and outer radii of the ring, respectively. The resulting watermark is written as

$$W_m(\omega) = M(\omega)e^{\,\mathrm{i}\theta_\omega}, \qquad \theta_{-\omega} = -\theta_\omega, \tag{2}$$

where θ_ω denotes the phase at ω. Together with the symmetric ring mask, this antisymmetry makes W_m Hermitian, ensuring that its inverse Fourier transform is real-valued while retaining the complete watermark pattern for detection (Ci et al., 2024; Lee & Cho, 2025).

4.2 Semantics-Preserving Watermark Embedding

The previous section describes how IRIS constructs an image-specific watermark. Because the watermark is derived from the visual representation of the generated image, IRIS first completes an unwatermarked generation to obtain a preview from which the watermark is derived. It then restarts from a late latent and embeds the derived watermark during the final few denoising steps.

A unique challenge of in-generation watermarking is determining **when to embed the watermark**. The generation process is dynamic and continuously alters the latent through denoising. Our intuition is that the remaining denoising steps may alter a watermark embedded before the latent has sufficiently stabilized. To examine this effect, we measure the normalized correlation between the one-step clean estimate $\hat{z}_0(i)$ at DDIM sampling step i and the final latent z_0 after 50 DDIM sampling steps. As shown in Fig. 2 (a), the latent becomes highly aligned with the final latent in the last few steps, motivating embedding from a late step onward, which we define as the start step i_0.

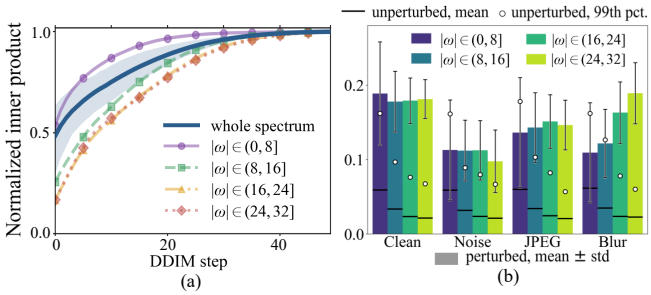

Figure 2: (a) Latent convergence toward the final latent. (b) Watermark recoverability through the VAE.

Another question is **where to embed the watermark**. We consider two factors: detectability and imperceptibility, which are often in tension. We examine this tension by focusing on the VAE decoding–encoding round trip under common image distortions. Specifically, we perturb different frequency bands of channel 0 of a clean latent from the Stable Diffusion 2.1-base VAE[1], decode it into an image, optionally apply JPEG compression (Q=25), noise (σ=0.1), or blur (σ=3), and re-encode the image for watermark detection. As shown in Fig. 2 (b), the low-frequency band ($|\omega| \in (0, 8]$) exhibits a high null (unwatermarked) response, resulting in poor separation between watermarked and unwatermarked signals, whereas the high-frequency band ($|\omega| \in (24, 32]$) is less robust to noise. We additionally observe that perturbing the low-frequency band causes the largest fidelity loss. Therefore, we place the watermark in a mid-frequency band of the latent.

[1]https://huggingface.co/Manojb/stable-diffusion-2-1-base

Let Z_i denote the Fourier coefficients of the latent channel at DDIM sampling step i. For $\omega \in \mathcal{R}$, we embed W_m by:

$$Z_i(\omega) \leftarrow (1 - \lambda_i) Z_i(\omega) + \lambda_i \gamma_i W_m(\omega), \tag{3}$$

where γ_i scales the unit-magnitude W_m to the root mean square (RMS) magnitude within $\mathcal{R}$, and λ_i decreases linearly from λ at the start step i_0 toward zero at the last step T.

4.3 Watermark Detection

Alignment score. It is highly likely that the observed Fourier coefficients $\hat{Z}_t$ will not be identical to the original Z_t due to our embedding method and the DDIM inversion error. To detect the presence of an embedded watermark, we measure the alignment between the expected watermark W_m and the observed Fourier coefficients $\hat{Z}_i$ using their normalized correlation (Solachidis & Pitas, 2001), where a higher alignment score indicates stronger evidence that $\hat{Z}_t$ matches W_m. This metric is different from the L1 distance used in previous work and is essential for forgery resistance (Section 5.4, Appendix B.8). For complex-valued ring coefficients, we use the Hermitian inner product $\langle W_m, \hat{Z}_t \rangle = \sum_{\omega \in \mathcal{R}} W_m(\omega)^* \, \hat{Z}_t(\omega)$. Let $\|\cdot\|_2$ denote the standard Euclidean norm. The alignment score is defined as

Algorithm 1. Watermark embedding of IRIS.

Require: Prompt c, secret key k, start step i_0, initial strength λ

$z_{i_0}, z_T \leftarrow$ DDIM sampling with c
$\bar{\mathbf{e}} \leftarrow$ visual representation of $\mathcal{D}(z_T)$ (Sec. 4.1)
$m \leftarrow \mathrm{SimHash}(\bar{\mathbf{e}}; k)$ (Eq. 1)
$W_m \leftarrow \mathrm{Gen}(m, k)$ (Eq. 2)
for $i = i_0, \ldots, T-1$ **do** ▷ restart from z_{i_0}
  Embed W_m into z_i with λ_i (Eq. 3)
  $z_{i+1} \leftarrow \mathrm{DDIM}(z_i, c)$
end for
return $x \leftarrow \mathcal{D}(z_T)$

$$s_t(m) = \frac{\left|\langle W_m, \hat{Z}_t \rangle\right|}{\|W_m\|_2 \, \|\hat{Z}_t\|_2}, \tag{4}$$

Watermark signals across inverted latents. An important question for IRIS is which inverted latents to use for watermark detection. Most previous work uses the latent at a symmetric inversion step for this task. In contrast, we find watermark signals persist across multiple latents obtained over a sequence of inversion steps. To examine this phenomenon, we embed a perturbation W with strength λ=0.2 at DDIM sampling step i=45 and measure its normalized correlation with the corresponding latent at each inversion step. As shown in Fig. 3, the detection signal continues to strengthen through the 20th DDIM inversion step, while the signal from unwatermarked images remains close to zero. Motivated by this observation, we perform D steps of DDIM inversion on the query image using an empty prompt and average the normalized correlation scores over these D inverted latents. For a given semantic code m, its detection score is $s(m) = \frac{1}{D} \sum_{t=1}^{D} s_t(m)$.

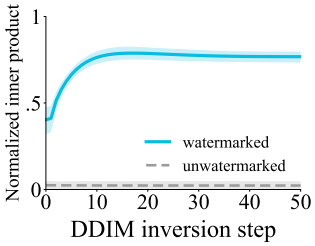

Figure 3: Watermark readout across inversion steps.

Watermark detection algorithm. To further enhance verification resilience to common image distortions, we construct a local candidate set of semantic codes during detection. Since SimHash preserves semantic similarity through code similarity, common image distortions are likely to flip bits closest to the SimHash partition boundaries or cause small bit drift. Given a long enough code length, code drift under image distortions remains far from the range of forged images. We therefore construct the candidate set $\mathcal{M}(\hat{m})$ by first flipping the L bits closest to the SimHash partition boundaries and supplementing it with neighbour codes within distance d. These two parameters provide a finer control over the verification resilience. Due to the relationship between the code and the pattern, in the candidate, at most one can reach a high alignment scores but the others remain low, the resulting

Algorithm 2. Watermark detection of IRIS.

Require: Image x, key k, inversion steps D, threshold τ

$\hat{m} \leftarrow \mathrm{SimHash}(x)$ (Eq. 1); build candidate set $\mathcal{M}(\hat{m})$
$\{\hat{Z}_t\}_{t=1}^{D} \leftarrow$ DDIM inversion of x with an empty prompt
for all $m \in \mathcal{M}(\hat{m})$ **do**
  $W_m \leftarrow \mathrm{Gen}(m, k)$; $s(m) \leftarrow \frac{1}{D} \sum_{t=1}^{D} s_t(m)$ (Eq. 4)
end for
$\rho^\star \leftarrow \max_m s(m) \,/\, \mathrm{median}_m \, s(m)$
return $\mathbb{I}[\rho^\star > \tau]$

detection statistic is the ratio between the highest and median candidate scores, denoted by ρ^*. A valid watermark is detected when $\rho^\star > \tau$.

Proposition 1 (Visual-semantic binding rejects forgeries that alter the semantics.) *Let x_s be a watermarked image and x_f a forged image that carries the watermark of x_s, and let α be the angle between their visual representations. Then*

$$\Pr[x_f \text{ passes detection}] \;\le\; \Pr\left[\mathrm{Bin}(\ell, \alpha/\pi) \le L\right] \;+\; \mathrm{FPR}(\tau), \tag{5}$$

where ℓ is the code length, L the number of low-margin bits flipped when building the candidate set, and $\mathrm{FPR}(\tau)$ the false-positive rate of the detector at threshold τ.

The first term is the chance that the semantic code of x_s is included in the candidate set built from x_f, which is small when the two images are semantically far apart. The second term says that, once the source code is not a candidate, the forged image is treated like an unwatermarked one. Based on Proposition 1, given a semantic code length and candidate set parameters, we can estimate a forgery success rate. For example, given $\ell = 24$, $L = 7$ and the measured cross-prompt angle $\alpha = 88.5°$, the first term is at most 0.038; a finer count that requires the differing bits to land on the L low-margin positions gives 1.9×10^{-4}, predicting an attack success rate close to the FPR.

5 Experiments

5.1 Experimental Setup

Settings. Our main experiments are conducted on Stable Diffusion 2.1-base[2] with 50-step DDIM sampling and a guidance scale of 7.5. We generate watermarked images using prompts from three datasets: Stable-Diffusion-Prompts (SDP) (Gustavosta, 2022), DiffusionDB (Wang et al., 2023), and MS-COCO (Lin et al., 2014). Our comparative analysis includes four *content-agnostic* baselines: Tree-Ring (Wen et al., 2023), RingID (Ci et al., 2024), Gaussian Shading (GS) (Yang et al., 2024b), PRC (Gunn et al., 2025); and two *content-aware* methods: SEAL (Arabi et al., 2025) and SemBind (Zhang et al., 2026). All baselines use their default official configurations. IRIS embeds a watermark at Fourier-frequency band $(20, 28]$ of latent channel 0 starting from DDIM step i=45, with an initial strength λ_i=0.2. The watermark is derived from a 24-bit semantic code mapped from a representation extracted by CLIP ViT-B/32 (Radford et al., 2021). Its detection uses a candidate set constructed with parameters L=9 and r=4 and averages the watermark signal over D=20 inversion steps.



Figure 4: Visual comparison of images generated using different watermarking methods (same prompt, same seed). PSNR, LPIPS are dataset-level.

Evaluation. To benchmark the *detectability*, we report the true positive rate (TPR) at a 1% false positive rate (TPR@1%FPR) and the area under the curve (AUC). To evaluate the *forgery resistance*, we use two black-box attacks introduced by Müller et al. (2025): Imprint-Forgery, which spoofs real MS-COCO images, and Reprompting, which regenerates new content from the inverted noise of a watermarked image. For both attacks, we use Stable Diffusion 1.5 [3] as the surrogate model. To evaluate **visual fidelity**, we measure PSNR and LPIPS (Zhang et al., 2018), instance-level metrics, between a watermarked image and its unwatermarked preview generated under the same setting except for watermarking, and Fréchet Inception Distance (FID) (Heusel et al., 2017), a distribution-level metric against MS-COCO images. The appendix provides more detailed experimental details including comparison with post-hoc methods.

[2]https://huggingface.co/stabilityai/stable-diffusion-2-1-base

[3]https://huggingface.co/stable-diffusion-v1-5/stable-diffusion-v1-5

5.2 Main Results: Detectability, Forgery-Resistance, and Fidelity

As shown in Table 1, for **detectability**, IRIS demonstrates high detectability with comparable AUC and TPR@1%FPR to the other content-aware methods. For **forgery resistance**, we report the average attack success rate (ASR) of Reprompt and Imprint-Forgery with 50/100/150 optimization steps for each method. IRIS reduces ASR to nearly the false positive rate, consistent with our analysis based on Proposition 1. All content-agnostic methods suffer from a severe ASR above 95%.

For content-aware methods, SEAL has an average ASR of 61.2%. SemBind applies a semantic mask to content-agnostic methods, reducing their forgery ASR accordingly. However, SemBind (+GS) and SemBind (+TR) still show a high average ASR compared to IRIS and slightly lower detectability than their base methods. For **fidelity**, as shown in Fig. 4, IRIS appears visually identical to its unwatermarked preview, with a PSNR of 30.9, an LPIPS of 0.03, and an FID of 23.7. In contrast, SEAL and SemBind produce watermarked images that look different from the unwatermarked image from which they derive the semantics for their watermarks. This helps explain why their semantic binding leaves room for forgery. Overall, IRIS achieves detectability, forgery-resistance, and high visual fidelity.

		Detectability	Forgery ASR ↓			Fidelity
	Method	T@1%F / AUC ↑	Imprint	Reprompt	Mean	FID ↓
Content-agnostic	Tree-Ring	1.000 / 1.000	0.970	0.956	0.963	25.1
	RingID	1.000 / 1.000	1.000	0.996	0.998	27.2
	Gaussian Shading	1.000 / 1.000	1.000	1.000	1.000	26.8
	PRC	1.000 / 1.000	0.987	0.982	0.984	23.9
Content-aware	SEAL	0.992 / 0.999	0.600	0.624	0.612	24.2
	SemBind (+TR)	0.963 / 0.996	0.000	0.226	0.113	24.7
	SemBind (+GS)	0.997 / 0.998	0.253	0.634	0.444	24.6
	IRIS (ours)	**1.000 / 1.000**	**0.020**	**0.022**	**0.021**	**23.7**

Table 1: Main results. T@1%F represents TPR@1%FPR.

5.3 Robustness to Distortions

We evaluate IRIS robustness to common distortions, including JPEG compression (Q=25), Gaussian blur (σ=3), Gaussian noise ($\sigma = 0.1$), brightness (factor 1.2), cropping (95%), rotation (5°), and regeneration (50 diffusion step). As shown in Tab. 2, IRIS maintains comparable detectability across these distortions. In depth, we decouple detection into two factors in IRIS's visual-semantic binding mechanism. We define the *oracle pattern detection rate* as the proportion of images for which the embedded pattern can be detected when the ground-truth code is provided to the detector, and the *code recovery rate* as the proportion of images whose ground-truth semantic codes are contained in their candidate sets constructed during detection. Across all distortions, oracle pattern detection stays high at 1.000, while code recovery slightly drops under distortions. This reduced detectability stems from lost visual-semantic binding.

Method	JPEG	Blur	Noise	Bright	Crop	Rot.	Regen.
Tree-Ring	0.860	0.738	0.888	1.000	1.000	1.000	0.962
Gaussian Shading	1.000	1.000	1.000	1.000	1.000	1.000	1.000
RingID	1.000	0.998	1.000	1.000	1.000	1.000	0.998
PRC	0.886	0.986	0.734	0.996	1.000	0.998	0.954
SemBind (+TR)	0.668	0.707	0.605	0.910	0.918	0.879	0.622
SemBind (+GS)	0.992	0.992	0.990	0.998	0.998	0.996	0.990
IRIS (ours)	**0.976**	**0.994**	**0.932**	**0.994**	**0.992**	**0.976**	**0.940**
oracle pattern	1.000	1.000	1.000	1.000	1.000	1.000	1.000
code recovery	0.992	0.994	0.982	0.994	0.992	0.976	0.938

Table 2: Detectability (T@1%F) under distortions.

5.4 Ablations, Generalization, and Computational Overhead

Ablation on design choices. We conduct ablation studies using the DiffusionDB dataset. As shown in Table 3, *without preprocessing* of the visual representation, IRIS detectability under common distortions drops heavily. Without *semantic binding*, which means the watermark is derived only from the key, IRIS T@1%F and AUC consistently achieve 1.0 across all scenarios, but suffer from high forgery ASR, a behaviour similar to a *content-agnostic watermark*. Replacing the normalized correlation with the L_1 distance of Tree-Ring, or dropping the max-over-median normalization, leaves detectability unchanged but raises the Imprint ASR from 0.02 to at least 0.97 or more: a transplanted pattern shifts the scores of all wrong candidates, and the ratio of normalized correlations is invariant to that shift (Appendix B.8). When reading the watermark signal *only at step 45* (symmetric to the step where the watermark starts to be embedded), detectability remains high in most scenarios but drops under noise. IRIS framework requires visual representations to derive semantic codes. We keep other components unchanged and instead use DINOv2 (Oquab et al., 2024) as the visual

Variant	Clean		Noise σ=0.1		JPEG Q=25		Blur σ=3		Forgery ASR ↓	
	T@1%F	AUC	T@1%F	AUC	T@1%F	AUC	T@1%F	AUC	Imprint	Reprompt
IRIS (default)	**1.000**	**1.000**	**0.932**	**0.970**	**0.976**	**0.994**	**0.994**	**0.997**	**0.020**	**0.022**
w/o preprocessing	0.988	0.994	0.560	0.783	0.488	0.751	0.186	0.614	0.010	0.008
w/o semantic binding	1.000	1.000	1.000	1.000	1.000	1.000	1.000	1.000	0.677	1.000
single signal at step 45	0.994	0.998	0.862	0.947	0.956	0.982	0.968	0.983	0.017	0.010
CLIP replaced with DINOv2	0.988	0.993	0.888	0.965	0.968	0.988	0.958	0.976	0.010	0.022
candidate set r=3, L=7	0.994	0.997	0.900	0.954	0.964	0.985	0.968	0.980	0.017	0.010

Table 3: IRIS performance in ablation studies on key design choices of IRIS.

encoder. The representations perform comparably, indicating IRIS is not sensitive to the choice of visual encoder. Visual encoders provide image semantics, but the IRIS framework implements the visual-semantic binding. We further test a candidate set r=3, L=7; performance is comparable with a very slight reduction in resilience, indicating that forgery resistance mainly benefits from the visual-semantic binding, and that the candidate set provides finer control over the binding.

Extension to flow-matching models. Although we motivate the design with LDMs, flow-matching models use a similar generation mechanism: they progressively transform noise into an image by solving an ODE. Table 4 shows the results of extending IRIS to two flow-matching models. IRIS gains comparative performance and higher fidelity. This indicates that IRIS is a general watermarking framework. For details, see Appendix D.

Model	T@1%F	PSNR	Imprint	Reprompt
SD3.5-medium	1.000	41.0	0.020	0.006
FLUX.2-klein	0.997	41.0	0.033	0.006

Table 4: IRIS on other models.

Two-layer deployment. IRIS works at very late sampling steps to minimize impact on the generated content. In contrast, baselines work at the initial noise, which determines the content of the generated results. We explore a two-layer deployment that uses the initial noise from baselines and then applies IRIS at the late sampling steps of their generation. As shown in Table 5, IRIS and initial noise-based watermarks can coexist and be reliably detected. This indicates that hierarchical watermarks can be constructed for advanced provenance in the future.

Method	Clean		JPEG Q=25		Noise σ=0.1		Brightness 1.2		
	Base	IRIS	Base	IRIS	Base	IRIS	Base	IRIS	PSNR
IRIS + TR	1.000	1.000	0.830	0.960	0.920	0.750	1.000	0.980	33.5
IRIS + GS	1.000	1.000	1.000	0.970	1.000	0.760	1.000	1.000	33.9
IRIS + PRC	1.000	1.000	0.870	1.000	0.640	0.860	1.000	0.990	32.0
IRIS + RingID	1.000	1.000	1.000	0.980	1.000	0.870	1.000	0.980	32.5

Table 5: Watermark detectability (T@1%F) and fidelity.

Computational overhead. Table 6 reports the end-to-end runtime for in-generation watermarking, dominated by UNet steps. IRIS has a runtime close to Gaussian Shading (2.49 vs. 2.46 s). SEAL is the most expensive at 6.33 s (2.6× the baseline), requiring two full sampling passes for embedding and inversion to the initial noise (150 UNet steps in total), as well as BLIP (Li et al., 2022) captioning during both stages. SemBind+GS takes 3.76 s (52.8% more), due to the additional 50 UNet steps for generating an auxiliary image.

Method	UNet steps	per image (s)		
	emb. / det.	embed	detect	total (s)
Gaussian Shading	50 / 50	1.29 (1.0×)	1.17	2.46
SEAL	100 / 50	3.89 (3.0×)	2.44	6.33
SemBind+GS	100 / 50	2.60 (2.0×)	1.16	3.76
IRIS (ours)	**55 / 20**	**1.69** (1.3×)	**0.80**	**2.49**

Table 6: Per-image wall-clock cost.

6 Conclusion

IRIS is a novel in-generation watermarking framework whose watermark is derived from the image it authenticates, achieving detectability, forgery resistance, and high fidelity simultaneously. Our comprehensive evaluation yields five key insights. (1) Augmenting images yields distortion-robust SimHash codes. (2) Embedding a perturbation in the latent during the last few steps can preserve image fidelity compared to not embedding. (3) The pattern embedded in the latent space during diffusion generation is persistent, and the strongest signal may not appear at a symmetric inversion step but at intermediate inversion steps. (4) Through SimHash semantic codes, the code gradually drifts elegantly under distortions, allowing distance-based local candidates to adjust the tightness of binding. (5) A two-stage watermark embedding at different diffusion sampling steps can maintain both embedded patterns. Our findings encourage exploration of the full capacity of hiding information in the diffusion generation process. We hope the discussion helps further exploration of watermarking methods that are more secure and reliable for AI-generated image provenance.

AI USE STATEMENT

During the preparation of this manuscript, we only employed large language models as auxiliary tools for non-substantive tasks. We retain full intellectual ownership of the research design, conceptual framework, experimental analysis, and scientific conclusions. Specifically, we strictly limited LLM assistance to the following specific tasks: polishing text fluency, checking grammatical consistency, and debugging sandbox execution scripts; assisting in literature searches to identify relevant prior work, without generating fictitious citations; aggregating source links for public datasets to accelerate data collection; and paraphrasing task instructions for the instruction generalization evaluation set to ensure diverse linguistic phrasing while preserving strict task requirements. All core methodologies, architectural designs, and insights presented in this work are entirely independent of LLM generative outputs.

REFERENCES

Anthropic. How claude's text watermark works. https://www.anthropic.com/news/claude-text-watermark, August 2026. Accessed: 2026-09-14.

Kasra Arabi, R Teal Witter, Chinmay Hegde, and Niv Cohen. Seal: Semantic aware image watermarking. In *Proceedings of the IEEE/CVF International Conference on Computer Vision*, pp. 16196–16205, 2025.

James Betker, Gabriel Goh, Li Jing, Tim Brooks, Jianfeng Wang, Linjie Li, Long Ouyang, Juntang Zhuang, Joyce Lee, Yufei Guo, et al. Improving image generation with better captions. Technical report, OpenAI, 2023. https://cdn.openai.com/papers/dall-e-3.pdf.

C2PA. Content credentials: C2PA technical specification, version 2.4. https://spec.c2pa.org/specifications/specifications/2.4/specs/C2PA_Specification.html, April 2026.

Moses S Charikar. Similarity estimation techniques from rounding algorithms. In *Proceedings of the thiry-fourth annual ACM symposium on Theory of computing*, pp. 380–388, 2002.

Hai Ci, Pei Yang, Yiren Song, and Mike Zheng Shou. Ringid: Rethinking tree-ring watermarking for enhanced multi-key identification. In *European conference on computer vision*, pp. 338–354. Springer, 2024.

Ingemar Cox, Matthew Miller, Jeffrey Bloom, Jessica Fridrich, and Ton Kalker. *Digital Watermarking and Steganography*. Morgan Kaufmann, 2 edition, 2007.

Patrick Esser, Sumith Kulal, Andreas Blattmann, Rahim Entezari, Jonas Müller, Harry Saini, Yam Levi, Dominik Lorenz, Axel Sauer, and Robin Rombach. Scaling rectified flow transformers for high-resolution image synthesis. In *International Conference on Machine Learning (ICML)*, 2024.

European Union. Regulation (EU) 2024/1689 laying down harmonised rules on artificial intelligence. https://eur-lex.europa.eu/eli/reg/2024/1689/oj, 2024.

Pierre Fernandez, Guillaume Couairon, Hervé Jégou, Matthijs Douze, and Teddy Furon. The stable signature: Rooting watermarks in latent diffusion models. In *Proceedings of the IEEE/CVF International Conference on Computer Vision*, pp. 22466–22477, 2023.

Sam Gunn, Xuandong Zhao, and Dawn Song. An undetectable watermark for generative image models. In *The Thirteenth International Conference on Learning Representations*, 2025. URL https://openreview.net/forum?id=jlhBFm7T2J.

Gustavosta. Stable diffusion prompts. https://huggingface.co/datasets/Gustavosta/Stable-Diffusion-Prompts, 2022. Accessed: 2026-07-28.

Martin Heusel, Hubert Ramsauer, Thomas Unterthiner, Bernhard Nessler, and Sepp Hochreiter. Gans trained by a two time-scale update rule converge to a local nash equilibrium. *Advances in neural information processing systems*, 30, 2017.

Jonathan Ho, Ajay Jain, and Pieter Abbeel. Denoising diffusion probabilistic models. In *Advances in Neural Information Processing Systems*, volume 33, 2020.

Seongmin Hong, Kyeonghyun Lee, Suh Yoon Jeon, Hyewon Bae, and Se Young Chun. On exact inversion of dpm-solvers. In *2024 IEEE/CVF Conference on Computer Vision and Pattern Recognition (CVPR)*, pp. 7069–7078. IEEE, 2024.

Huayang Huang, Yu Wu, and Qian Wang. Robin: Robust and invisible watermarks for diffusion models with adversarial optimization. *Advances in Neural Information Processing Systems*, 37: 3937–3963, 2024.

Anubhav Jain, Yuya Kobayashi, Naoki Murata, Yuhta Takida, Takashi Shibuya, Yuki Mitsufuji, Niv Cohen, Nasir Memon, and Julian Togelius. Forging and removing latent-noise diffusion watermarks using a single image, 2025. URL https://arxiv.org/abs/2504.20111.

Sung Ju Lee and Nam Ik Cho. Semantic watermarking reinvented: Enhancing robustness and generation quality with fourier integrity. In *2025 IEEE/CVF International Conference on Computer Vision (ICCV)*, pp. 18759–18769. IEEE, 2025.

Junnan Li, Dongxu Li, Caiming Xiong, and Steven Hoi. Blip: Bootstrapping language-image pre-training for unified vision-language understanding and generation. In *International conference on machine learning*, pp. 12888–12900. PMLR, 2022.

Kecen Li, Zhicong Huang, Xinwen Hou, and Cheng Hong. Gaussmarker: Robust dual-domain watermark for diffusion models. In *Forty-second International Conference on Machine Learning*, 2025. URL https://openreview.net/forum?id=m7Mx14cxv8.

Tsung-Yi Lin, Michael Maire, Serge Belongie, James Hays, Pietro Perona, Deva Ramanan, Piotr Dollár, and C Lawrence Zitnick. Microsoft coco: Common objects in context. In *European conference on computer vision*, pp. 740–755. Springer, 2014.

Yaron Lipman, Ricky T. Q. Chen, Heli Ben-Hamu, Maximilian Nickel, and Matt Le. Flow matching for generative modeling. In *International Conference on Learning Representations*, 2023.

Andreas Müller, Denis Lukovnikov, Jonas Thietke, Asja Fischer, and Erwin Quiring. Black-box forgery attacks on semantic watermarks for diffusion models. In *2025 IEEE/CVF Conference on Computer Vision and Pattern Recognition (CVPR)*, pp. 20937–20946. IEEE Computer Society, 2025.

OpenAI. Verify OpenAI-generated content. https://openai.com/research/verify/, 2026. Accessed: 2026-09-14.

Maxime Oquab, Timothée Darcet, Théo Moutakanni, Huy V. Vo, Marc Szafraniec, Vasil Khalidov, Pierre Fernandez, Daniel HAZIZA, Francisco Massa, Alaaeldin El-Nouby, Mido Assran, Nicolas Ballas, Wojciech Galuba, Russell Howes, Po-Yao Huang, Shang-Wen Li, Ishan Misra, Michael Rabbat, Vasu Sharma, Gabriel Synnaeve, Hu Xu, Herve Jegou, Julien Mairal, Patrick Labatut, Armand Joulin, and Piotr Bojanowski. DINOv2: Learning robust visual features without supervision. *Transactions on Machine Learning Research*, 2024. ISSN 2835-8856. URL https://openreview.net/forum?id=a68SUt6zFt. Featured Certification.

Alec Radford, Jong Wook Kim, Chris Hallacy, Aditya Ramesh, Gabriel Goh, Sandhini Agarwal, Girish Sastry, Amanda Askell, Pamela Mishkin, Jack Clark, Gretchen Krueger, and Ilya Sutskever. Learning transferable visual models from natural language supervision. In Marina Meila and Tong Zhang (eds.), *Proceedings of the 38th International Conference on Machine Learning*, volume 139 of *Proceedings of Machine Learning Research*, pp. 8748–8763. PMLR, 18–24 Jul 2021. URL https://proceedings.mlr.press/v139/radford21a.html.

Robin Rombach, Andreas Blattmann, Dominik Lorenz, Patrick Esser, and Björn Ommer. High-resolution image synthesis with latent diffusion models. In *Proceedings of the IEEE/CVF conference on computer vision and pattern recognition*, pp. 10684–10695, 2022.

Patrick Schramowski, Manuel Brack, Björn Deiseroth, and Kristian Kersting. Safe latent diffusion: Mitigating inappropriate degeneration in diffusion models. In *2023 IEEE/CVF Conference on Computer Vision and Pattern Recognition (CVPR)*, pp. 22522–22531. IEEE, 2023.

Vassilios Solachidis and Loannis Pitas. Circularly symmetric watermark embedding in 2-d dft domain. *IEEE transactions on image processing*, 10(11):1741–1753, 2001.

Jiaming Song, Chenlin Meng, and Stefano Ermon. Denoising diffusion implicit models. In *International Conference on Learning Representations*, 2021.

Christian Szegedy, Vincent Vanhoucke, Sergey Ioffe, Jon Shlens, and Zbigniew Wojna. Rethinking the inception architecture for computer vision. In *Proceedings of the IEEE conference on computer vision and pattern recognition*, pp. 2818–2826, 2016.

Zijie J Wang, Evan Montoya, David Munechika, Haoyang Yang, Benjamin Hoover, and Duen Horng Chau. Diffusiondb: A large-scale prompt gallery dataset for text-to-image generative models. In *Proceedings of the 61st annual meeting of the association for computational linguistics (volume 1: Long papers)*, pp. 893–911, 2023.

Yuxin Wen, John Kirchenbauer, Jonas Geiping, and Tom Goldstein. Tree-rings watermarks: Invisible fingerprints for diffusion images. *Advances in Neural Information Processing Systems*, 36: 58047–58063, 2023.

Jindong Yang, Han Fang, Weiming Zhang, Nenghai Yu, and Kejiang Chen. T2smark: balancing robustness and diversity in noise-as-watermark for diffusion models. *Advances in Neural Information Processing Systems*, 38:118642–118668, 2026.

Pei Yang, Hai Ci, Yiren Song, and Mike Zheng Shou. Can simple averaging defeat modern watermarks? *Advances in Neural Information Processing Systems*, 37:56644–56673, 2024a.

Zijin Yang, Kai Zeng, Kejiang Chen, Han Fang, Weiming Zhang, and Nenghai Yu. Gaussian shading: Provable performance-lossless image watermarking for diffusion models. In *Proceedings of the IEEE/CVF Conference on Computer Vision and Pattern Recognition*, pp. 12162–12171, 2024b.

Richard Zhang, Phillip Isola, Alexei A Efros, Eli Shechtman, and Oliver Wang. The unreasonable effectiveness of deep features as a perceptual metric. In *2018 IEEE/CVF conference on computer vision and pattern recognition*, pp. 586–595. IEEE, 2018.

Xin Zhang, Zijin Yang, Kejiang Chen, Linfeng Ma, Weiming Zhang, and Nenghai Yu. Sembind: Binding diffusion watermarks to semantics against black-box forgery attacks. In *Forty-third International Conference on Machine Learning*, 2026.

Xuandong Zhao, Kexun Zhang, Zihao Su, Saastha Vasan, Ilya Grishchenko, Christopher Kruegel, Giovanni Vigna, Yu-Xiang Wang, and Lei Li. Invisible image watermarks are provably removable using generative ai. *Advances in neural information processing systems*, 37:8643–8672, 2024.

Jiren Zhu, Russell Kaplan, Justin Johnson, and Li Fei-Fei. Hidden: Hiding data with deep networks. In *European conference on computer vision*, pp. 682–697. Springer, 2018.

A Proof of Proposition 1

Write m_s for the code embedded in x_s, $\hat{m}_f$ for the code recovered from x_f at detection, and $\mathcal{M}(\hat{m}_f)$ for the candidate set built from $\hat{m}_f$. The forged image can pass in only two ways: either m_s is inside the candidate set, so its own pattern is tested and found; or m_s is outside, and some *other* candidate happens to score high. Therefore,

$$\Pr[\text{pass}] \;\le\; \underbrace{\Pr[m_s \in \mathcal{M}(\hat{m}_f)]}_{\text{Step 1}} \;+\; \underbrace{\Pr[\text{pass} \mid m_s \notin \mathcal{M}(\hat{m}_f)]}_{\text{Step 2}}. \tag{6}$$

We bound each piece.

Step 1: when can m_s be a candidate? Every candidate is obtained from $\hat{m}_f$ by flipping at most r bits, or by flipping some of L chosen bits. Either way, it differs from $\hat{m}_f$ in at most $\max(r, L) = L$ positions. So if m_s is a candidate, then the Hamming distance H between m_s and $\hat{m}_f$ is at most L. Consider how H is distributed. Each bit of a SimHash code is the sign of a random projection $\mathbf{r}_j^\top(\bar{\mathbf{e}} - \boldsymbol{\mu})$. For two vectors at angle α, a random hyperplane separates them with probability α/π (a classical SimHash fact of Charikar (2002)), and the ℓ hyperplanes are independent. Hence $H \sim \mathrm{Bin}(\ell, \alpha/\pi)$, and

$$\Pr[m_s \in \mathcal{M}(\hat{m}_f)] \;\le\; \Pr[H \le L]. \tag{7}$$

This is the first term of equation 5. Now suppose m_s is not a candidate.

Step 2: a non-candidate source looks unwatermarked. Each candidate pattern W_m is produced by a keyed generator seeded with m and the secret key. Without the key, and for any $m \neq m_s$, the phases of W_m are fresh pseudorandom values that have nothing to do with the phases of W_{m_s} sitting in the forged latent. So from the detector's point of view, testing the candidates of $\mathcal{M}(\hat{m}_f)$ against x_f is the same experiment as testing them against an image that carries no candidate pattern at all. The probability of passing in that experiment is, by definition, the false-positive rate $\mathrm{FPR}(\tau)$ at which we calibrated τ. Adding the two steps gives Eq.5. □

Numbers. With $\ell = 24$, $L = 7$, $\alpha = 88.5°$ ($\alpha/\pi = 0.49$), $\Pr[H \le 7] = 0.038$. If the flipped positions are as likely to land on the L low-margin bits as anywhere else, the chance that m_s is a candidate is $\Pr[H \le r] + \sum_{h=r+1}^{L} \Pr[H = h]\binom{L}{h}/\binom{\ell}{h} = 1.9 \times 10^{-4}$, the same order as the 6.4×10^{-4} we measure (Table 8); the measured value is a little larger because low-margin bits are indeed the ones most likely to flip. For images that keep the source semantics ($\alpha \approx 8°$ under JPEG or blur), the same term is close to 1, which makes detection resilient.

B Experimental Details

Here, we provide more details on our experimental setup, including the configuration of IRIS and competing methods, and the evaluation protocols used throughout our experiments.

B.1 Image Generation and Evaluation Setup

Image generation. Unless otherwise specified, we generate images with Stable Diffusion 2.1-base at a resolution of 512×512, corresponding to a latent size of $4\times64\times64$. We use DDIM sampling with 50 steps and a classifier-free guidance scale of 7.5, with the UNet and VAE in half precision. We evaluate on three prompt sets: Stable-Diffusion-Prompts (SDP), DiffusionDB, and MS-COCO captions. For each set, we sample 500 prompts and assign a fixed seed to each prompt. To make a fair comparison between watermarked and unwatermarked generation, each pair uses the same prompt, seed, and initial noise. For IRIS, watermark embedding begins only during the final five DDIM steps, leaving the preceding 45 steps unchanged.

Detection thresholds. We use a separate pool of 1,500 unwatermarked images for threshold calibration, consisting of 500 images from each of the three prompt sets and generated under the same protocol. We calibrate each detector to a 1% false-positive rate (FPR) on this pool. AUC is computed using the watermarked evaluation images and the same unwatermarked pool.

B.2 IRIS Configuration

Visual semantic readout. We use CLIP ViT-B/32 to extract a 512-dimensional visual representation. As described in Sec. 4, we form $K = 8$ augmented views of the image; each view is independently subjected to JPEG compression, additive Gaussian noise, and Gaussian blur with probability 0.8 each. We sample JPEG quality uniformly from $[25, 70)$, the standard deviation of Gaussian noise from $[0, 18]$ on the 0–255 pixel scale, and the blur standard deviation from $[0, 3]$. Augmentations use a fixed per-image random seed, making the semantic readout deterministic. We then canonicalize the image by normalizing its mean luminance to 0.5, applying Gaussian blur with $\sigma = 6$, and JPEG re-encoding at quality 25. We finally average the representations of K augmented views. The averaged representation is ℓ_2-normalized before semantic-code mapping.

Visual semantic code. We use a 24-bit SimHash for the visual semantic code. The random hyperplanes are sampled independently from $\mathcal{N}(0, I)$ using a generator seeded by the secret key. We center the representation using the mean embedding $\boldsymbol{\mu}$ of $6{,}000$ calibration images, with $2{,}000$ images from each prompt set. These calibration images are disjoint from the 500 evaluation prompts. We retain the signed SimHash projections as bit margins for constructing the candidate set during detection. The same hyperplanes, $\boldsymbol{\mu}$, and secret key are used for embedding and detection.

Watermark embedding. For watermark embedding, we use channel 0 of the latent and place the pattern in eight one-pixel-wide annuli $(20 + k, 21 + k]$, $k = 0, \ldots, 7$, in the centred 2-D Fourier plane, i.e., the carrier occupies the band of frequency radii $(20, 28]$ of the 64×64 latent channel, split into eight rings of unit width. The resulting carrier contains $1{,}196$ Fourier coefficients. Given a semantic code m, we generate the phase-only pattern using a pseudorandom generator seeded by $\mathrm{HMAC}(\mathrm{key}, m, i_0)$ and impose Hermitian symmetry. Watermark embedding starts at DDIM step $i_0 = 45$ and continues through the final five DDIM sampling steps. At each step, the watermark magnitude is matched independently to the root-mean-square magnitude of the latent coefficients in each annulus. The interpolation strength decreases linearly from $\lambda_{45} = 0.2$ to $\lambda_{49} = 0.04$, following $\lambda_i = 0.2(50 - i)/5$.

Watermark detection. At detection, we encode the query image with the VAE and perform $D = 20$ null-prompt DDIM inversion steps. We compute the alignment score over all 20 inverted latent states, as defined in Sec. 4.3. To account for small errors in semantic-code recovery, the candidate set includes all codes within Hamming distance $r = 4$ of the recovered code, plus all codes obtained by flipping arbitrary subsets of its $L = 9$ lowest-margin bits; this is slightly larger than the theoretical bound but still achieve superior performance empirically. We compute the normalized correlation for each candidate and use the max-over-median statistic $\rho^\star$ for detection. We calibrate its threshold at 1% FPR using the unwatermarked pool described above.

B.3 Baselines

We compare IRIS with Tree-Ring, RingID, Gaussian Shading, PRC, SEAL, and SemBind. All methods use the same Stable Diffusion 2.1-base backbone, prompts, seeds, and a 50-step DDIM sampler. We use the official implementations and default configurations. Because different methods define detection statistics differently, we keep their original detectors but calibrate their decision thresholds on the same $1{,}500$ unwatermarked images at 1% FPR.

For **Tree-Ring**[4], we use the official "ring" key: ten concentric one-pixel rings of radii 1 to 10 in the centred Fourier plane of channel 0 of the initial noise, each ring carrying one Fourier coefficient of a Gaussian sample drawn from a fixed key seed; the key replaces the corresponding coefficients of the initial noise, and detection uses the negative L_1 distance between the key and the inverted noise inside the ring mask after 50-step DDIM inversion. **RingID**[5] uses its official multi-ring construction and detector with the default radii and channels of the authors' verification script. For **Gaussian Shading**[6], we embed a fixed 256-bit message with channel factor 1 and 8×8 blocks (each bit repeated over 64 latent positions) and measure detection by the bit accuracy of the recovered message. PRC[7] uses the official pseudo-random code construction, key generation and decoder. For **SEAL**[8], $k = 1024$, $b = 7$, and $\tau = 2.3$. We obtain a BLIP-2 (Flan-T5-XL) caption from the unwatermarked preview and use the caption embedder released with the authors' code, along with their SimHash implementation, to derive its watermark. **SemBind**[9] is evaluated on top of both Tree-Ring and Gaussian Shading using the authors' semantic masker (DINOv2-Giant backbone) and the mask ratio the authors use for these two schemes.

Zero-bit evaluation protocol. Gaussian Shading, RingID, and PRC are multi-bit schemes; we evaluate them as zero-bit detectors (watermark present or absent) for comparison. For Gaussian Shading, a single fixed 256-bit message is embedded in every image, and the score is the bit accuracy of the recovered message against it; for RingID, one key of the official codebook is fixed for all

[4] https://github.com/YuxinWenRick/tree-ring-watermark
[5] https://github.com/showlab/RingID
[6] https://github.com/bsmhmmlf/Gaussian-Shading
[7] https://github.com/XuandongZhao/PRC-Watermark
[8] https://github.com/Kasraarabi/SEAL
[9] https://github.com/XinZhang1999/SemBind

images, and the score is the negative L_1 distance to that key's pattern, leaving out the nearest-key identification step; PRC's official DETECT is a zero-bit test; therefore, we directly use it. In every case, we set the decision threshold to the empirical 1%-FPR for the same false-positive budget.

B.4 FORGERY ATTACKS

We evaluate two black-box forgery attacks, Imprint-Forgery and Reprompting, following Müller et al. (2025). In both attacks, the adversary uses Stable Diffusion 1.5 as a surrogate model and has no access to the watermark owner's model or secret key.

Imprint-Forgery. We use 100 watermarked source images generated from SDP prompts and pair them with 100 unrelated target images randomly sampled from MS-COCO and resized to 512×512. We use the same source–target pairs for all watermarking methods. For each pair, the attacker optimizes the target's latent representation so that its inverted initial noise approaches that of the watermarked source. Following the official implementation of Müller et al. (2025), we perform inversion with a 10-step differentiable DDIM process using the surrogate model, optimize with Adam for 150 steps using a learning rate of 0.01, evaluate the forged image with the victim detector at 50, 100 and 150 steps, and report attack success at these budgets. Following the attacker-keeps-the-best protocol, an attack is successful at budget B if any checkpoint up to B passes the victim detector. The reported Imprint attack success rate (ASR) averages across the three budgets. Table 7 reports ASR at each budget.

Reprompting. For Reprompting, the attacker first reconstructs the initial noise of a watermarked source image using 50-step DDIM inversion with the surrogate model. Starting from this noise, we generate a new image using a different prompt with 50 denoising steps and guidance scale 7.5. Attack prompts are taken from the Inappropriate Image Prompts (I2P) dataset (Schramowski et al., 2023), with the i-th prompt paired with the i-th watermarked source image. An attack is successful if the regenerated image is accepted by the victim detector. We report ASR over all source images.

		Imprint-Forgery ASR ↓ at budget B			
	Method	B=50	B=100	B=150	Mean
Content-agnostic	Tree-Ring	0.910	1.000	1.000	0.970
	RingID	1.000	1.000	1.000	1.000
	Gaussian Shading	1.000	1.000	1.000	1.000
	PRC	0.980	0.980	1.000	0.987
Content-aware	SEAL	0.230	0.700	0.870	0.600
	SemBind (+TR)	0.000	0.000	0.000	0.000
	SemBind (+GS)	0.070	0.240	0.450	0.253
	IRIS (ours)	**0.010**	**0.020**	**0.030**	**0.020**

Table 7: Imprint-Forgery attack success rate by optimization budget. An attack succeeds at budget B if any checkpoint up to B passes the detector's clean 1%-FPR threshold. The Imprint column of Table 1 is the mean over the three budgets.

B.5 VISUAL FIDELITY

We evaluate instance-level fidelity using PSNR, SSIM, and LPIPS with an AlexNet backbone. Each watermarked image is compared with the unwatermarked image generated from the same prompt, seed, and initial noise. This pairing isolates the visual change introduced by watermark embedding from the inherent stochastic variation of diffusion generation.

We additionally report FID against real images. Specifically, we use the first $5{,}000$ captions from MS-COCO val2017 to generate $5{,}000$ watermarked images and compute FID against their corresponding MS-COCO images, resized to 512×512. FID is computed using the $2{,}048$-dimensional pool3 features of Inception-v3 (Szegedy et al., 2016), following the standard protocol of Heusel et al. (2017) as implemented in `pytorch-fid`[10].

[10]https://github.com/mseitzer/pytorch-fid

For two-layer deployment, where IRIS is applied to initially noise-watermarked images, we evaluate PSNR on the watermarked image with only those watermarking methods.

B.6 Ablation Settings

Unless otherwise stated, all ablations keep the configuration above unchanged. Detectability is evaluated on the 500 DiffusionDB pairs. Forgery resistance is evaluated with the same attack protocol and samples as in the main experiments.

Without semantic conditioning. To isolate the contribution of conditioning the watermark on visual semantics, we replace the pattern derived from a specific image with a single fixed keyed pattern shared across images. Since detection now considers only one pattern, the candidate set contains a single element; we directly threshold the normalized correlation at its 99th percentile on the unwatermarked pool. For the forgery experiments, we evaluate each attack image against the pattern carried by its source image.

Without multi-step averaging. We replace the average over $D = 20$ inverted states with a single state. We consider two choices: the state after five inversion steps, corresponding to generation step 45 and therefore the watermark embedding window, and the deepest state after 20 inversion steps, corresponding to generation step 30.

Without preprocessing. We remove the preprocessing pipeline and derive the semantic code directly from a single raw CLIP embedding. We recompute the calibration mean $\boldsymbol{\mu}$ from raw embeddings of the same $6{,}000$ calibration images. We regenerate the watermarked images and unwatermarked calibration pool and rerun both forgery attacks under this setting.

L_1 distance instead of normalized correlation. To examine the role of the correlation-based score, we replace the normalized correlation by the mean absolute

$$d_m = \frac{1}{D} \sum_{t=1}^{D} \frac{1}{|W|} \left\| \hat{Z}_t - \kappa_t W_m \right\|_1 , \tag{8}$$

where κ_t is a fixed scale for inversion step t: W_m is phase-only (unit magnitude) whereas the coefficients of $\hat{Z}_t$ have a magnitude that depends on t, so the pattern must be scaled before the distance is taken. We set κ_t to the root-mean-square magnitude of the carrier coefficients at step t, averaged over the unwatermarked calibration images; the scales $\kappa_1, \ldots, \kappa_D$ are computed once and held fixed for all query images and candidate codes, which gives the detector a fixed-magnitude key in the sense of Tree-Ring. The candidate set is unchanged. Since a smaller distance means a better match, the max-over-median statistic becomes $\mathrm{median}_m d_m / \min_m d_m$, which is again greater than or equal to 1 and increases with the evidence for a watermark; its threshold is the 1%-FPR quantile on the same unwatermarked pool.

Semantic encoder. We study whether the choice of visual encoder affects IRIS by replacing CLIP ViT-B/32 with DINOv2-base or SigLIP-base-patch16-224. For DINOv2, we use the CLS token as the image representation. Each encoder uses its own calibration mean $\boldsymbol{\mu}$ and detection threshold, calibrated at 1% FPR.

B.7 Analysis of Visual Semantic Representations

The visual-semantic representation in IRIS serves two purposes: it should distinguish images with different visual semantics, while remaining stable when the same image undergoes common distortions. We therefore examine both *encoder dependence* and *representation stability*. Unless otherwise specified, detection is evaluated on the 500 DiffusionDB pairs, both clean and under JPEG compression (Q=25), Gaussian noise (σ=0.1), and Gaussian blur (σ=3).

Dependence on the visual encoder. We first examine whether IRIS's visual-semantic binding depends on a particular pretrained visual encoder by replacing CLIP ViT-B/32 with DINOv2-B and SigLIP-B while keeping the rest of the pipeline unchanged. Each encoder uses its own calibration

mean $\boldsymbol{\mu}$ and detection threshold. Besides end-to-end detectability, we report *collision*, the probability that the candidate region of one protocol code contains the code of an unrelated image; *margin*, the median absolute SimHash projection $|p_i|$; and *code recovery*, the fraction of watermarked images whose embedding-time code remains inside the candidate set at detection. As shown in Table 8, all three encoders achieve similar detectability and very low collision rates (3.9×10^{-4}–6.8×10^{-4}), despite noticeable differences in projection margin and distortion sensitivity. DINOv2 yields the largest margin (0.597) without uniformly improving detection, while SigLIP attains the highest code recovery (0.984) but is more sensitive to Gaussian noise. Overall, these results indicate that different pretrained visual representations can provide a suitable semantic space for code construction, and that IRIS's visual-semantic binding is not tied to CLIP specifically.

	TPR@1%FPR ↑				Code separability		Code recovery ↑
Encoder	Clean	JPEG	Noise	Blur	Collision ↓	Margin ↑	
CLIP ViT-B/32	1.000	0.976	0.994	0.986	6.4×10^{-4}	0.370	0.976
DINOv2-B	0.988	0.968	0.958	0.978	3.9×10^{-4}	0.597	0.970
SigLIP-B	0.990	0.966	0.892	0.992	6.8×10^{-4}	0.394	0.984

Table 8: Effect of the visual encoder on code separability, recovery, and watermark detectability.

Effect of representation preprocessing. We next isolate the role of the canonicalization and augmentation pipeline by examining semantic-code stability under JPEG compression. We compare the default pipeline with a *raw* variant that directly extracts a single CLIP ViT-B/32 representation without preprocessing. As shown in Table 9, preprocessing substantially reduces code drift as compression becomes stronger: without preprocessing, the average Hamming distance increases from 1.26 bits at $Q{=}90$ to 3.62 bits at $Q{=}20$, while code recovery drops from 0.994 to 0.746; with preprocessing, the Hamming distance remains below one bit down to $Q{=}30$, with code recovery staying at 1.000. These results show that robustness primarily come from stabilizing its representation before semantic-code mapping, allowing common distortions to induce only local code changes that can be absorbed by the candidate set without weakening visual-semantic binding.

B.8 Analysis of the Detection Statistic

The detector in Section 4 scores every candidate code $m \in \mathcal{M}$ by the normalized correlation s_m between its pattern W_m and the inverted bands $\hat{Z}_t$, and decides with the ratio $\rho^\star = \max_m s_m / \mathrm{median}_m s_m$. This rule folds in two design choices: the score (a normalized correlation) and the normalization (a ratio to the candidate median). This appendix separates them by scoring the same persisted bands with six statistics (Table 10), looking at the resulting score distributions (Fig. 5), and then explaining why only one of the six resists forgery.

Statistics. *Normalized correlation* is $s_m = \frac{1}{D}\sum_t |\langle W_m, \hat{Z}_t\rangle| / (\|W_m\|\|\hat{Z}_t\|)$. L_1 *(plain)* is the distance used by Tree-Ring, $d_m = \frac{1}{D}\sum_t \frac{1}{|W|}\|\hat{Z}_t - \kappa_t W_m\|_1$, with a fixed per-state scale κ_t equal to the mean root-mean-square magnitude of the carrier band over the unwatermarked pool (a one-off calibration, like the fixed magnitude of the Tree-Ring key). L_1 *(energy-matched)* rescales each observed band to the pattern's magnitude before the distance, $\|\hat{Z}_t\,\mathrm{rms}(W_m)/\mathrm{rms}(\hat{Z}_t) - W_m\|_1$, mirroring the energy matching at injection. For each score the *raw* statistic is the best candidate ($\max_m s_m$ or $-\min_m d_m$) and the *ratio* statistic divides it by the candidate median ($\max_m s_m / \mathrm{median}_m s_m$ or $\mathrm{median}_m d_m / \min_m d_m$). In this experiment, we use the candidate set rule $r{=}3$, $L{=}7$ and detectability is measured on the 500 DiffusionDB pairs and the attack rates on the Imprint-Forgery checkpoints ($50/100/150$ steps, averaged) and the reprompted images of Table 3.

Empirical observations. All six statistics detect clean watermarks with TPR at or near 1 and remain robust to common distortions; the plain L_1 variants are even slightly stronger in several distortion settings. Their main difference appears under forgery. With $\rho^\star$, forged and reprompted images remain close to the unwatermarked distribution (median 5.25 versus 5.24 for unwatermarked images; Fig. 5), whereas the other statistics frequently cross their detection thresholds. Neither normalized correlation nor ratio normalization alone explains this behaviour: the raw normalized correlation yields Imprint and Reprompt ASRs of 0.973 and 0.960, respectively. In comparison, both L_1 ratio

Representation	Metric	JPEG quality							
		90	80	70	60	50	40	30	20
raw	Code recovery ↑	0.994	0.978	0.954	0.960	0.926	0.868	0.820	0.746
	Hamming distance ↓	1.26	1.87	2.02	2.18	2.53	3.03	3.23	3.62
preprocessed	Code recovery ↑	1.000	1.000	1.000	1.000	1.000	1.000	1.000	0.996
	Hamming distance ↓	0.37	0.48	0.62	0.71	0.80	0.83	0.95	1.54

Table 9: Semantic-code stability under JPEG quality w/wo preprocessing.

Statistic	Clean		Noise σ=0.1		JPEG Q=25		Blur σ=3		Forgery ASR ↓	
	T@1%F	AUC	T@1%F	AUC	T@1%F	AUC	T@1%F	AUC	Imprint	Reprompt
Normalized correlation, raw	1.000	1.000	0.916	0.980	0.988	0.998	0.998	1.000	0.973	0.960
Normalized correlation, ratio (ours)	0.994	0.997	0.902	0.954	0.964	0.985	0.968	0.980	0.010	0.010
L_1 (plain), raw	1.000	1.000	0.962	0.999	0.996	1.000	0.994	1.000	0.300	0.852
L_1 (plain), ratio	1.000	1.000	0.914	0.978	0.984	0.996	0.996	1.000	0.977	0.930
L_1 (energy-matched), raw	1.000	1.000	0.902	0.980	0.978	0.997	0.992	1.000	0.937	0.938
L_1 (energy-matched), ratio	1.000	1.000	0.916	0.980	0.984	0.997	0.994	1.000	0.970	0.942

Table 10: Detection-statistic ablation on the same inverted bands. Every statistic detects clean and distorted watermarks well; only the ratio of normalized correlations resists forgery.

variants still exhibit ASRs of at least 0.93. This shows that forgery resistance arises specifically from combining normalized correlation with candidate-wise ratio normalization.

Candidate scores under forgery. A forged or reprompted image carries the pattern issued for the *source* image, while its candidate set $\mathcal{M}$ is constructed from the semantic code of the *new* content. Across the attacked images considered here, the source code falls inside $\mathcal{M}$ for none of the 300 forged checkpoints and for one of the 500 reprompted images. Consequently, apart from this single case, none of the tested candidate patterns matches the transplanted pattern. The candidate scores therefore form a background distribution determined by the observed band rather than by a true pattern match. An attack may shift this background distribution, but without recovering the source code it cannot selectively increase the score of the correct candidate.

Why raw scores fail. A raw detector uses only the best candidate and is therefore sensitive to changes in the overall candidate-score distribution. Under Imprint-Forgery, for example, the median raw normalized-correlation score increases from 0.10 for unwatermarked images to 0.18 for forged images, causing the best-candidate statistic to cross its threshold even though no candidate corresponds to the transplanted pattern. This leads to an Imprint ASR of 0.973. Raw L_1 statistics exhibit the same general failure mode.

Why normalized-correlation ratios resist forgery. The ratio $\rho^\star = \max_m s_m / \operatorname{median}_m s_m$ measures whether one candidate stands out relative to the remaining candidates rather than whether all candidate scores become jointly larger. Normalized correlation further removes the overall magnitude of both the expected pattern and the observed band, making the relative candidate-score distribution substantially less sensitive to attack-induced changes in carrier energy. Thus, when no candidate matches the transplanted pattern, forgery tends to raise the candidate scores together without creating a distinctive maximum, and the ratio remains near its unwatermarked level. In contrast, L_1 distances remain sensitive to the coefficient-wise magnitude distribution of the observed band: changing this distribution can alter the spread of distances across candidates, so their best-to-median ratio may increase even without a matching candidate. This explains why both L_1 ratio variants remain vulnerable (Imprint ASR 0.977 and 0.970), whereas the normalized-correlation ratio reduces the Imprint ASR to 0.010.

Summary. These results identify three complementary ingredients for forgery resistance: a content-derived candidate set that excludes the transplanted source pattern when the visual semantics change; a relative statistic that tests whether one candidate stands out from the candidate background; and a normalized similarity measure whose relative scores are insensitive to changes in overall carrier magnitude. Together, these properties explain why $\rho^\star$ succeeds where the alternative statistics in Table 10 fail.

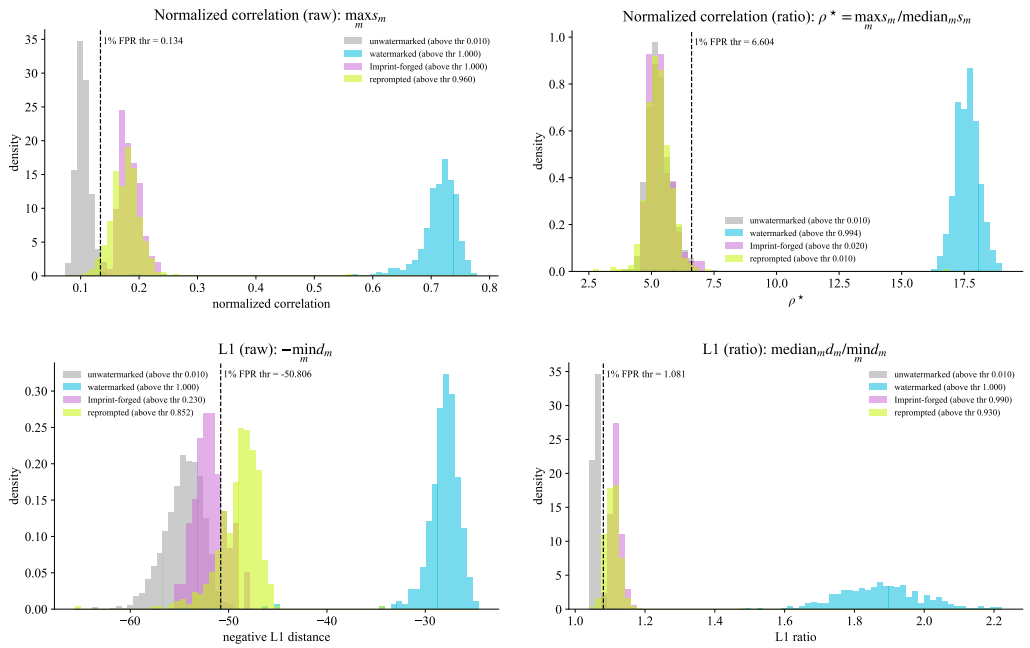

Figure 5: Score distributions of unwatermarked, watermarked, Imprint-forged (100 steps), and reprompted images for the raw (left) and ratio (right) statistics of the normalized correlation (top) and the plain L_1 distance (bottom); the dashed line is the 1%-FPR threshold. All four are separated from unwatermarked images; only $\rho^\star$ keeps the forged and reprompted images within the unwatermarked distribution.

C Comparison with Post-Hoc Methods

Although IRIS embeds its watermark only during the final few sampling steps, it remains fundamentally different from post-hoc watermarking. We therefore compare IRIS with Stable Signature (Fernandez et al., 2023) and DWT-DCT-SVD Cox et al. (2007)under the same clean and distorted evaluation settings. As shown in Table 11, IRIS achieves perfect clean detection and remains highly robust to JPEG compression, Gaussian noise, and diffusion regeneration. In contrast, the post-hoc methods degrade substantially under at least one of these transformations.

	Method	Clean	JPEG Q=25	Noise σ=0.1	Regen.	PSNR ↑
Post-poc	Stable Signature	1.000	0.906	0.672	0.006	29.5
	DWT-DCT-SVD	1.000	0.134	0.812	0.260	38.4
In-generation	Tree-Ring	1.000	0.860	0.888	0.962	–
	Gaussian Shading	1.000	1.000	1.000	1.000	–
	RingID	1.000	1.000	1.000	0.998	–
	PRC	1.000	0.886	0.734	0.954	–
	SemBind (+TR)	0.963	0.668	0.605	0.622	–
	SemBind (+GS)	0.997	0.992	0.990	0.990	–
	IRIS (ours)	**1.000**	**0.976**	**0.932**	**0.940**	**30.9**

Table 11: Robustness (T@1%F, 500 DiffusionDB images) and fidelity of post-hoc and in-generation watermarks. Regeneration re-noises the image to diffusion step t=50 and denoises it with SD 2.1. PSNR is measured against the unwatermarked image of the same prompt and seed; for initial-noise watermarks this comparison is not meaningful (–), because changing the initial noise changes the content of the generated image (9–14 dB), not only its details.

D Extension to Flow-Matching Models

IRIS operates on latent Fourier representations and inverted latent states, so extending it to flow-matching models requires only three backbone-specific choices: carrier channels, embedding steps, and inversion. The semantic code, keyed pattern, strength schedule, and detection statistic $\rho^\star$ remain unchanged. We evaluate Stable Diffusion 3.5-medium and FLUX.2-klein-4B at 512×512 with 50 Euler steps using their respective noise schedules and guidance scales of 4.5 and 4.0.

Channel selection. Because latent channels differ in how well their VAEs preserve Fourier perturbations, we screen each channel using a VAE encode–decode round trip on 100 images. We inject the deployed pattern on $(20, 28]$ and measure its *VAE keep ratio*, together with CLIP cosine and PSNR. Semantic similarity remains nearly unchanged across channels (CLIP cosine ≈ 0.997), while higher keep ratios generally trade off against pixel-level fidelity. Since all channels retain PSNR above 44 dB, we select those with the highest keep ratios. For SD3.5, channels 14 and 8 carry $(20, 24]$ and $(24, 28]$; for FLUX.2, channels 3, 8, 15, and 19 each carry a two-pixel-wide shell.

SD3.5-medium (16 latent channels)									
Channel	Keep ↑	CLIP cos	PSNR	Shells	Channel	Keep ↑	CLIP cos	PSNR	Shells
14	0.982	0.9971	47.3	$(20, 24]$	9	0.947	0.9976	50.0	–
8	0.979	0.9976	48.9	$(24, 28]$	11	0.944	0.9976	50.8	–
2	0.977	0.9974	48.6	–	1	0.941	0.9976	49.9	–
7	0.974	0.9972	48.0	–	0	0.937	0.9974	50.6	–
15	0.971	0.9968	47.6	–	12	0.937	0.9973	50.5	–
3	0.970	0.9974	48.6	–	13	0.924	0.9973	50.6	–
5	0.964	0.9972	49.0	–	4	0.920	0.9975	50.7	–
6	0.959	0.9971	48.5	–	10	0.910	0.9975	50.4	–

Table 12: Latent-channel screening for SD3.5-medium: VAE encode–decode keep ratio of a keyed pattern on the carrier band $(20, 28]$ (mean over 100 images, strength λ=0.2), CLIP cosine and PSNR between the decoded perturbed and clean images, and the shells assigned to the channel (– = not used). Channels are sorted by keep ratio.

FLUX.2-klein (32 latent channels)									
Channel	Keep ↑	CLIP cos	PSNR	Shells	Channel	Keep ↑	CLIP cos	PSNR	Shells
3	0.920	0.9968	44.0	$(20, 22]$	30	0.645	0.9970	54.5	–
8	0.918	0.9968	44.4	$(22, 24]$	21	0.644	0.9970	54.5	–
15	0.872	0.9970	50.2	$(24, 26]$	27	0.641	0.9970	54.7	–
19	0.870	0.9971	50.3	$(26, 28]$	28	0.632	0.9970	54.7	–
29	0.815	0.9970	52.6	–	25	0.622	0.9970	54.3	–
1	0.813	0.9971	51.9	–	4	0.615	0.9969	54.7	–
26	0.812	0.9970	52.0	–	20	0.614	0.9970	54.5	–
5	0.810	0.9969	51.8	–	12	0.612	0.9970	54.5	–
31	0.805	0.9970	52.0	–	16	0.596	0.9970	54.7	–
24	0.791	0.9970	52.9	–	18	0.585	0.9970	54.6	–
2	0.783	0.9970	53.4	–	17	0.553	0.9969	54.9	–
11	0.709	0.9970	52.7	–	23	0.553	0.9971	54.7	–
6	0.709	0.9970	53.9	–	0	0.531	0.9970	54.6	–
14	0.708	0.9970	53.9	–	10	0.510	0.9970	54.6	–
22	0.688	0.9971	54.3	–	9	0.506	0.9971	54.8	–
7	0.661	0.9969	54.5	–	13	0.496	0.9970	55.4	–

Table 13: Latent-channel screening for FLUX.2-klein: VAE encode–decode keep ratio of a keyed pattern on the carrier band $(20, 28]$ (mean over 100 images, strength λ=0.2), CLIP cosine and PSNR between the decoded perturbed and clean images, and the shells assigned to the channel (– = not used). Channels are sorted by keep ratio.

Embedding and inversion. Both backbones use deterministic Euler sampling over 50 sigma levels. As in SD 2.1, IRIS first generates an unwatermarked preview, restarts from step i_0=45, and embeds the pattern during the final five sampling steps with $\lambda_i = 0.2(50 - i)/5$. The semantic code and pattern generator are shared across all three backbones. Both flow-matching models achieve 41.0 dB PSNR between watermarked and unwatermarked images, compared with 30.9 dB on SD 2.1. At detection, we approximately invert the image with the reverse Euler update for D=20 steps and apply the same candidate-based $\rho^\star$ detector.

Results. Table 14 compares all three backbones under the same $r{=}3$, $L{=}7$ candidate set, with SD 2.1 re-evaluated under this protocol. IRIS retains near-perfect clean detectability and low forgery ASRs on both flow-matching models while achieving 41.0 dB PSNR. Distortion robustness remains high on SD3.5 but is lower on FLUX.2, particularly under JPEG compression (0.874) and Gaussian noise (0.794), consistent with its lower VAE keep ratios. These results indicate that the IRIS framework generalizes beyond DDIM-based diffusion models, although watermark robustness remains affected by backbone-specific latent characteristics.

		SD 2.1-base		SD3.5-medium		FLUX.2-klein	
		T@1%F	AUC	T@1%F	AUC	T@1%F	AUC
Clean	SDP	1.000	0.997	1.000	1.000	0.990	0.997
	DiffusionDB	1.000	0.997	1.000	1.000	1.000	1.000
	MS-COCO	1.000	0.999	1.000	1.000	1.000	1.000
	Mean	1.000	0.998	1.000	1.000	0.997	0.999
Distortion	JPEG $Q{=}25$	0.964	0.985	0.940	0.980	0.874	0.956
	Noise $\sigma{=}0.1$	0.902	0.954	0.868	0.952	0.794	0.924
	Blur $\sigma{=}3$	0.968	0.980	0.994	0.995	0.934	0.970
Forgery ASR ↓	Imprint-Forgery	0.023		0.020		0.033	
	Reprompting	0.010		0.006		0.006	
Fidelity	PSNR ↑	30.9		41.0		41.0	

Table 14: IRIS on flow-matching backbones. Clean results use 500 images per prompt set and distortion results use 500 DiffusionDB pairs. Imprint-Forgery ASR is averaged over the $50/100/150$-step budgets. SD 2.1 is re-evaluated under the same protocol for reference.

E Multi-bit Watermarking via Two-Layer Deployment

We further examine whether IRIS can coexist with multi-bit watermarks carried by the initial noise, enabling a two-layer deployment in which the base watermark conveys a message while IRIS provides visual-semantic binding. We consider Gaussian Shading with a 256-bit message and RingID with an 11-bit message, and apply IRIS from sampling step $i_0{=}45$ to 200 SDP images for each method. As shown in Table 15, adding IRIS does not affect recovery of the base messages: both methods retain a bit accuracy of 1.000 and a base-watermark detection rate of 1.000, while IRIS is simultaneously detected in every two-layer image. For RingID, the correct key is also identified among all 2^{11} keys for every image. The two-layer images retain PSNRs of 34.0 dB and 32.5 dB relative to their base-watermarked counterparts for Gaussian Shading and RingID, respectively. These results show that late-step IRIS embedding can add visual-semantic binding without disrupting information already encoded in the initial noise, supporting the use of the two watermark layers for complementary purposes.

		Base bit accuracy ↑		Base detected		IRIS detected	
Base watermark	Message	Base only	+ IRIS	Base only	+ IRIS	Base only	+ IRIS
Gaussian Shading	256 bits	1.000	1.000	1.000	1.000	0.030	1.000
RingID	11 bits (2^{11} keys)	1.000	1.000	1.000	1.000	0.010	1.000

Table 15: Multi-bit accuracy and detection under two-layer deployment (200 SDP images).

F Adaptive Attacks

We consider adaptive adversaries that target the visual-semantic binding of IRIS. The visual encoder and the complete algorithm are public, while the secret key, and hence the keyed SimHash hyperplanes and HMAC seeds, remain unknown. We study two attacks: whether surrogate keys can approximate the semantic-code mapping, and whether an attacker can directly optimize a forged image toward the source representation.

Threat model. Following Kerckhoffs' principle, the adversary knows the encoder, canonicalization and augmentation pipeline, code length $\ell{=}24$, candidate rule ($r{=}4$, $L{=}9$), carrier region, embedding

step i_0=45, and detection statistic. The provider's exact calibration centre $\boldsymbol{\mu}$ is not available, but we do not rely on it for secrecy and allow the adversary to estimate it from independent images. The adversary has black-box access to watermarked images and uses Stable Diffusion 1.5 as a surrogate generator.

F.1 Surrogate Key

Setup. The adversary runs the semantic-code pipeline with five independently sampled surrogate keys $K_1, \ldots, K_5$ and estimates the calibration centre using $1{,}500$ independent unwatermarked images. On $1{,}000$ watermarked images, we measure (i) the Hamming distance between true-key and surrogate-key codes, (ii) whether the true code falls inside the $r4L9$ candidate set constructed under a surrogate key, and (iii) whether surrogate keys can predict code collisions between image pairs.

	True key	K_1	K_2	K_3	K_4	K_5
True key	0	12.01	12.19	12.20	12.17	11.68
K_1	12.01	0	11.62	11.71	11.87	11.98
K_2	12.19	11.62	0	11.92	12.21	11.65
K_3	12.20	11.71	11.92	0	11.91	12.14
K_4	12.17	11.87	12.21	11.91	0	12.04
K_5	11.68	11.98	11.65	12.14	12.04	0
True code in the surrogate candidate set	–	0.000	0.000	0.000	0.000	0.001

Table 16: Semantic-code agreement under surrogate keys ($1{,}000$ watermarked images).

Surrogate keys do not recover the semantic codes. As shown in Table 16, independently sampled keys differ from the true key by 11.7–12.2 of 24 bits on average, close to the 12-bit expectation of independent binary codes. The true code falls inside a surrogate candidate set for at most 0.1% of images. In contrast, replacing the provider's centre $\boldsymbol{\mu}$ with the attacker's estimate $\hat{\boldsymbol{\mu}}$ changes the true-key code by only 0.42 bits on average, with $\cos(\boldsymbol{\mu}, \hat{\boldsymbol{\mu}}) = 0.9995$. Thus, secrecy comes from the keyed hyperplane orientations rather than the calibration centre; knowledge of the public encoder alone does not recover the semantic-code mapping.

CLIP angle θ	Pairs	Collision rate (true key)	Hamming (true key)	Agreement	Precision	Recall
$[0°, 5°)$	568	1.000	0.8	0.999	1.000	0.999
$[5°, 10°)$	219	0.941	1.7	0.911	0.941	0.965
$[10°, 15°)$	88	0.693	3.2	0.600	0.702	0.738
$[15°, 20°)$	182	0.429	4.4	0.558	0.485	0.444
$[20°, 30°)$	624	0.090	6.8	0.842	0.183	0.218
$[30°, 45°)$	1626	0.007	10.3	0.985	0.046	0.067
$[45°, 90°)$	1893	0.000	12.4	1.000	–	–

Table 17: Surrogate-key prediction of code collisions across CLIP-angle ranges ($5{,}200$ image pairs).

Collision prediction is reliable only at geometric extremes. A forger does not necessarily need to recover the code itself, but may instead try to predict whether the source code will fall inside the candidate set constructed from the forged image. Table 17 shows that surrogate keys predict this event almost perfectly for very similar or very dissimilar image pairs: pairs below $5°$ nearly always collide, whereas pairs above $30°$ rarely do. Near the decision boundary ($10°$–$20°$), however, prediction becomes substantially less reliable because the particular hyperplanes separating a pair depend on the secret key. The security of IRIS therefore does not rely on treating the semantic codes themselves as an additional secret. Rather, semantically different images are unlikely to map into the same candidate region (Proposition 1), while the watermark pattern associated with a code is independently derived from $\mathrm{HMAC}(\mathrm{key}, m, i_0)$ and cannot be reproduced without the key.

F.2 Gray-Box Embedding-Matching Forgery

Threat model. In a gray-box setting, we give the adversary gradient access to the public visual encoder and to its own differentiable approximation of the canonicalization pipeline. The secret key and its derived hyperplanes remain unknown.

Setup. We construct a gray-box attack based on Imprint-Forgery, in which the attacker jointly optimizes the surrogate inversion objective and the cosine distance between the forged and source representations:

$$\mathcal{L}(\delta) = \frac{\|\mathcal{I}_A(x_t+\delta) - \hat{z}_s\|^2}{\|\mathcal{I}_A(x_t) - \hat{z}_s\|^2} \; + \; \lambda \left(1 - \cos(\mathbf{e}_{\text{canon}}(x_t+\delta), \mathbf{e}_{\text{canon}}(x_s))\right), \tag{9}$$

subject to $\|\delta\|_\infty \le \epsilon$. Here, $\mathcal{I}_A$ is the differentiable 10-step inversion of the SD 1.5 surrogate. The attacker's canonicalization surrogate includes luminance normalization, Gaussian blur, and expectation over random noise-and-blur views; JPEG re-encoding is omitted from the differentiable attack path. We evaluate 50 source–target pairs with $\epsilon{=}16/255$, $\lambda{=}1$, and 150 PGD steps.

	Surrogate cosine	Canonical angle θ	Hamming to m_s	$\rho^\star$ (final)	PSNR to x_t
After 150 steps	0.937–0.966	31°–47° (mean 39°)	6–13 (min over steps 6–12)	4.9–5.8	27.1 dB

Table 18: Gray-box embedding-matching attack on source–target pairs ($\epsilon{=}16/255$, 150 PGD steps).

Embedding matching does not recover the source code. The attack successfully increases similarity in its own differentiable surrogate, reaching a cosine of 0.94–0.97 with the source. However, under the provider's full preprocessing pipeline, the resulting angle remains 31°–47°, and the forged code stays 6–13 bits from the source code, remaining outside the candidate set throughout the evaluated attack. Accordingly, all final detection statistics remain below threshold ($\rho^\star = 4.9$–5.8 versus a threshold of 6.62). The perturbation also reduces PSNR to 27.1 dB. These results suggest that the randomized and non-differentiable preprocessing introduced for distortion robustness also makes direct embedding-matching attacks substantially more difficult.

F.3 Pixel-Level Spoofing by Overlay

An adaptive adversary may try to spoof the visual-semantic binding directly in pixel space by overlaying a watermarked source image onto unrelated target content. We examine how much source content is required before the composite maps close enough to the source semantic code to enter its candidate region.

Setup. We use the 100 SDP sources from the Imprint-Forgery experiment and their paired MS-COCO targets. The source is pasted at the centre with edge ratio $r \in \{0.1, \ldots, 0.9\}$, corresponding to an area fraction of r^2. For each composite, we report its Hamming distance to the source code, whether its $r{=}4$, $L{=}9$ candidate set contains the source code, and its canonicalized CLIP angle to the source.

Overlay ratio r	0	0.1	0.2	0.3	0.4	0.5	0.6	0.7	0.8	0.9
Area of the source	0%	1%	4%	9%	16%	25%	36%	49%	64%	81%
Hamming distance (of 24)	12.4	12.4	12.2	11.7	10.9	10.1	9.2	8.1	7.0	5.8
Source code in candidate set	0.000	0.000	0.000	0.000	0.010	0.010	0.030	0.070	0.130	0.340
CLIP angle to the source (°)	51.1	50.4	49.0	46.5	43.8	40.5	36.8	33.1	29.0	22.8

Table 19: Semantic-code response to centred source overlays on 100 source–target pairs.

Candidate-set collisions remain rare until the pasted source dominates. Table 19 shows that the Hamming distance to the source code decreases steadily as the overlay ratio increases. Up to $r{=}0.5$, where the source covers 25% of the image area, the code remains far from the source code (10.1–12.4 bits) and the candidate-set hit rate is at most 1%. Even at $r{=}0.7$ (49% area), the average distance remains 8.1 bits and the source code enters the candidate set for only 7% of composites. At $r{=}0.9$, where the source covers 81% of the image and the target remains only as a surrounding frame, the distance drops to 5.8 bits and the hit rate rises to 34%. The accompanying decrease in CLIP angle indicates that the semantic representation gradually shifts toward the pasted source as its visual contribution increases. Figure 6 illustrates this gradual shift: as the pasted source occupies more of the image, the Hamming distance to its semantic code generally decreases, although the rate of change varies substantially across source–target pairs. Although the semantic code changes smoothly with increasing overlay, candidate-set collisions remain uncommon until the pasted source dominates the image.



Figure 6: Examples of centred overlay spoofing on four source–target pairs. Columns show the target, composites with increasing overlay ratio r, and the source; H denotes the Hamming distance between the composite and source semantic codes.

G Qualitative Examples

Figure 7 provides qualitative examples corresponding to the quantitative results in Tables 1 and 2. Clean watermarked images remain confidently detected under Gaussian noise and regeneration, where the original content is largely preserved. In contrast, Reprompting and Imprint-Forgery replace the image content and cause all examples to fall below the detection threshold. These examples illustrate the intended behavior of IRIS: the watermark remains verifiable under content-preserving transformations but is rejected when the visual semantics are replaced.

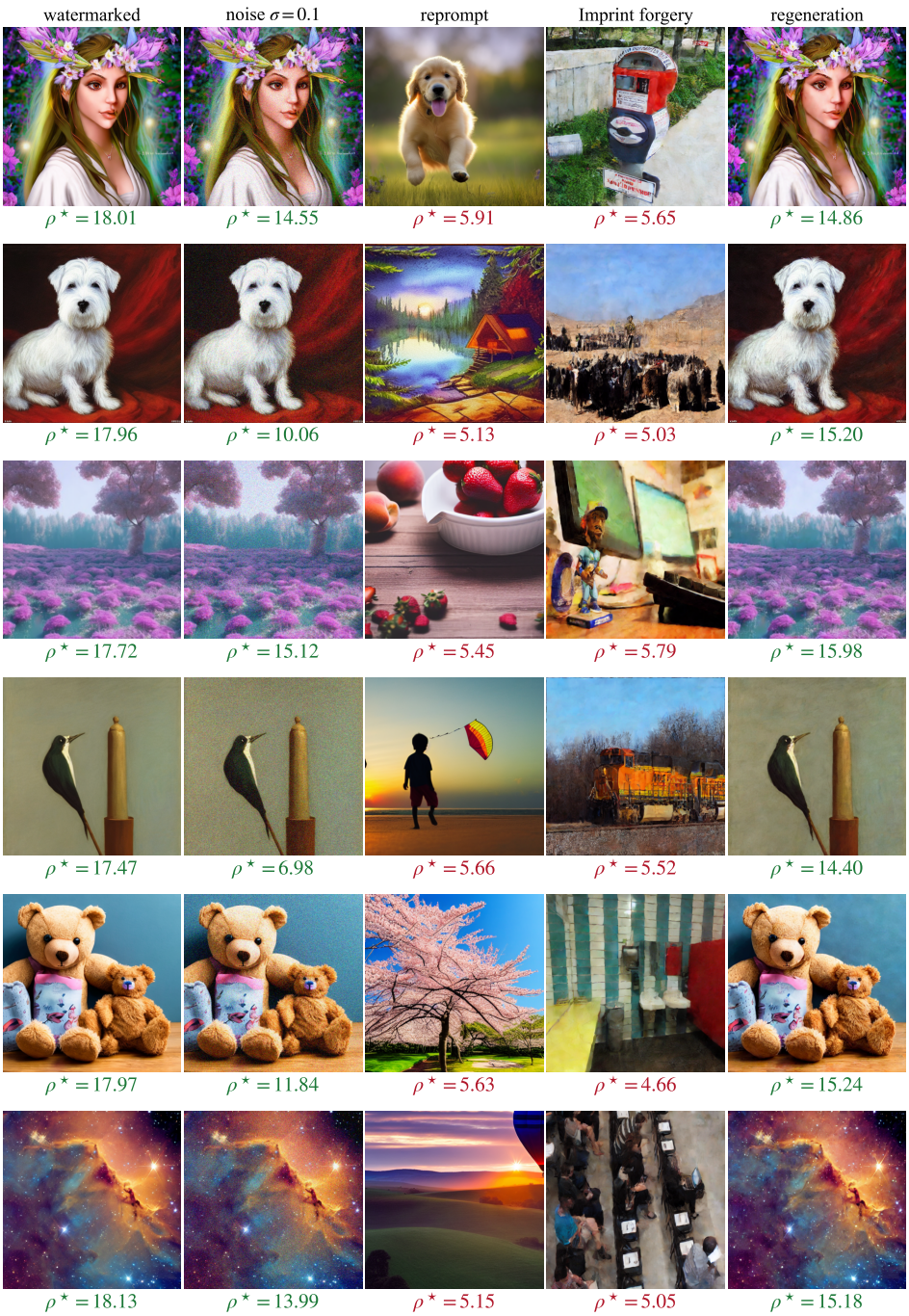

Figure 7: Qualitative detection examples under Gaussian noise, reprompting, imprint forgery, and regeneration. The value below each image is the IRIS detection statistic $\rho^\star$; green denotes acceptance at the 1%-FPR threshold and red denotes rejection.